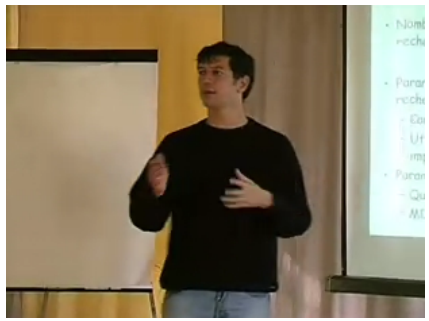
In-place Update of Suffix Array while Recoding Words

Matthias Gallé, Pierre Peterlongo, François Coste



Symbiose

INRIA

Prague Stringology Conference
September, 1st 2008

Motivation

- Grammatical Inference (on biological sequences) in the sense of Grammar-based codes*

*Kieffer & Yang 2000 “Grammar-based codes a new class of universal lossless source codes”

Motivation

- Grammatical Inference (on biological sequences) in the sense of Grammar-based codes*
- inspired by Sequitur†

*Kieffer & Yang 2000 “Grammar-based codes a new class of universal lossless source codes”

†Nevill-Manning, Witten “Compression and explanation using hierarchical grammars” 1997

Sketch of our GI algorithm

Sketch of our GI algorithm

0. Given a sequence...

ACGCATCTCCATCGCGCATATCATC

Sketch of our GI algorithm

1. select the “best” repeat

ACGCATCTCCATCGCGCATATCATC

⇓

ACGCATCTCCATCGCGCATATCATC

Sketch of our GI algorithm

2. replace it with a **new** symbol

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\qquad$ $M_1 \rightarrow CAT$

Sketch of our GI algorithm

... and so on...

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\quad M_1 \rightarrow CAT$

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\quad M_1 \rightarrow CAT$

Sketch of our GI algorithm

... and so on...

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\quad M_1 \rightarrow CAT$

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\quad M_1 \rightarrow CAT$

$\Downarrow$

ACGM_2TCM_2GCGM_1ATM_2 $\quad M_1 \rightarrow CAT$

$M_2 \rightarrow M_1 C$

Sketch of our GI algorithm

... and so on...

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGCATCTCCATCGCGCATATCATC

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\quad M_1 \rightarrow CAT$

$\Downarrow$

ACGM_1CTCM_1CGCGM_1ATM_1C $\quad M_1 \rightarrow CAT$

$\Downarrow$

ACGM_2TCM_2GCGM_1ATM_2 $\quad M_1 \rightarrow CAT$, $M_2 \rightarrow M_1C$

$\Downarrow$

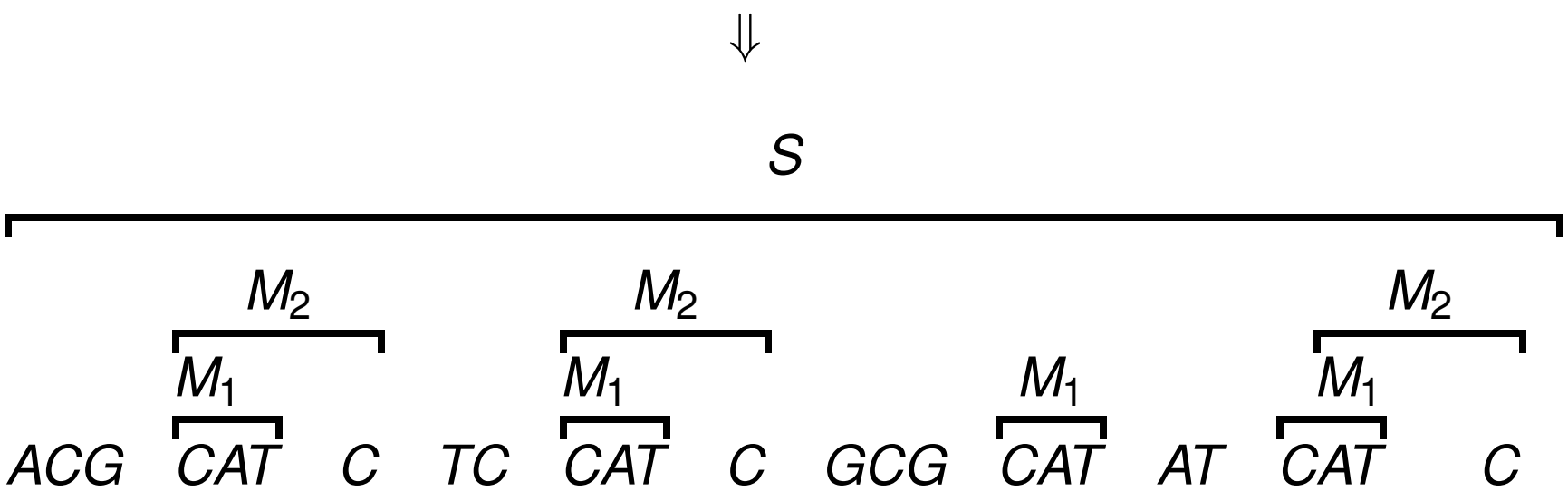
$S \rightarrow$ ACGTCM_2GCGM_1ATM_2
$M_1 \rightarrow CAT$
$M_2 \rightarrow M_1C$

Sketch of our GI algorithm

$S \rightarrow \mathrm{ACGTC}M_2\mathrm{GCG}M_1\mathrm{AT}M_2$
$M_1 \rightarrow CAT$
$M_2 \rightarrow M_1 C$

Sketch of our GI algorithm

$S \to \mathrm{ACGTC}M_2\mathrm{GCG}M_1\mathrm{AT}M_2$
$M_1 \to CAT$
$M_2 \to M_1 C$



Enhanced Suffix Array [Abouelhoda, Kurtz, et al 2004]

Enhanced Suffix Array [Abouelhoda, Kurtz, et al 2004]

ABRACADABRA $\rightarrow$ ABRACADABRA$

Enhanced Suffix Array [Abouelhoda, Kurtz, et al 2004]

ABRACADABRA → ABRACADABRA$

i	suffix
0	$
1	A$
2	ABRA$
3	ABRACADABRA$
4	ACADABRA$
5	ADABRA$
6	BRA$
7	BRACADABRA$
8	CADABRA$
9	DABRA$
10	RA$
11	RACADABRA$

Enhanced Suffix Array [Abouelhoda, Kurtz, et al 2004]

	sarr	+		+		=
i				sarr		suffix
0				11		$
1				10		A$
2				7		ABRA$
3				0		ABRACADABRA$
4				3		ACADABRA$
5				5		ADABRA$
6				8		BRA$
7				1		BRACADABRA$
‘ 8				4		CADABRA$
9				6		DABRA$
10				9		RA$
11				2		RACADABRA$

Enhanced Suffix Array [Abouelhoda, Kurtz, et al 2004]

	sarr	+	*lcp*	+		=
i			lcp		sarr	suffix
0			0		11	$
1			0		10	A$
2			1		7	ABRA$
3			4		0	ABRACADABRA$
4			1		3	ACADABRA$
5			1		5	ADABRA$
6			0		8	BRA$
7			3		1	BRACADABRA$
‘ 8			0		4	CADABRA$
9			0		6	DABRA$
10			0		9	RA$
11			2		2	RACADABRA$

Enhanced Suffix Array [Abouelhoda, Kurtz, et al 2004]

sarr + *lcp* + *isa* = *ESA*

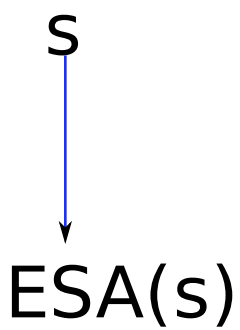
i	isa	lcp	sarr	suffix
0	3	0	11	$
1	7	0	10	A$
2	11	1	7	ABRA$
3	4	4	0	ABRACADABRA$
4	8	1	3	ACADABRA$
5	5	1	5	ADABRA$
6	9	0	8	BRA$
7	2	3	1	BRACADABRA$
‘ 8	6	0	4	CADABRA$
9	10	0	6	DABRA$
10	1	0	9	RA$
11	0	2	2	RACADABRA$

GI algorithm using ESA

GI algorithm using ESA

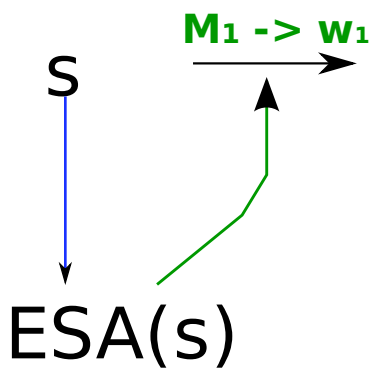
S

GI algorithm using ESA



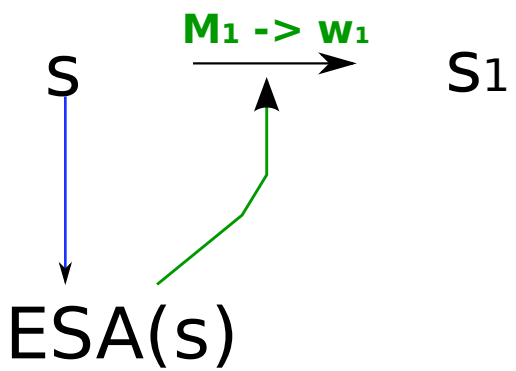
GI algorithm using ESA

Select “best” repeat



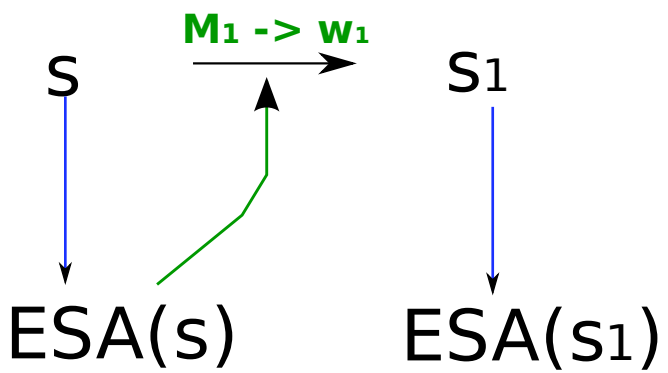
GI algorithm using ESA

Replace some (all) occurrences of the same repeat by a new character



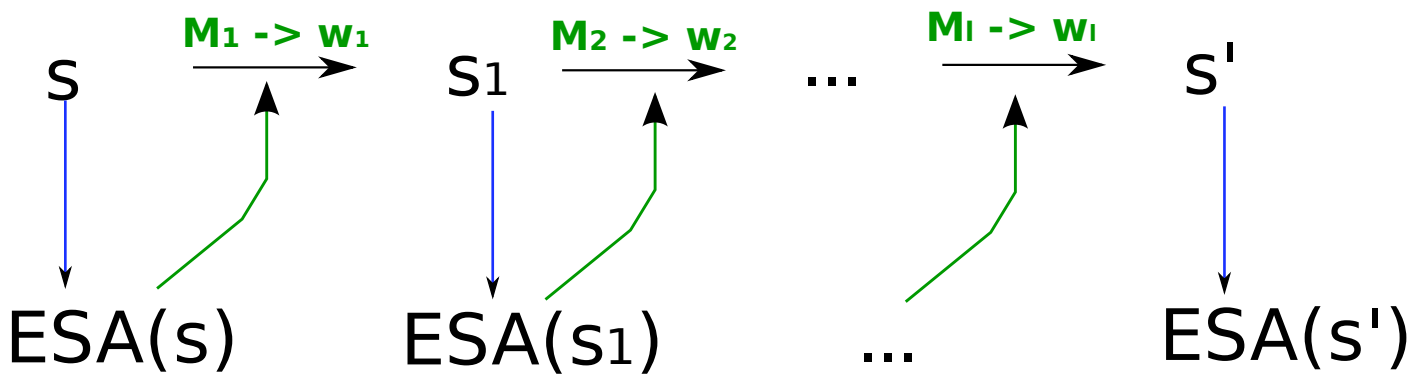
GI algorithm using ESA

Replace some (all) occurrences of the same repeat by a new character



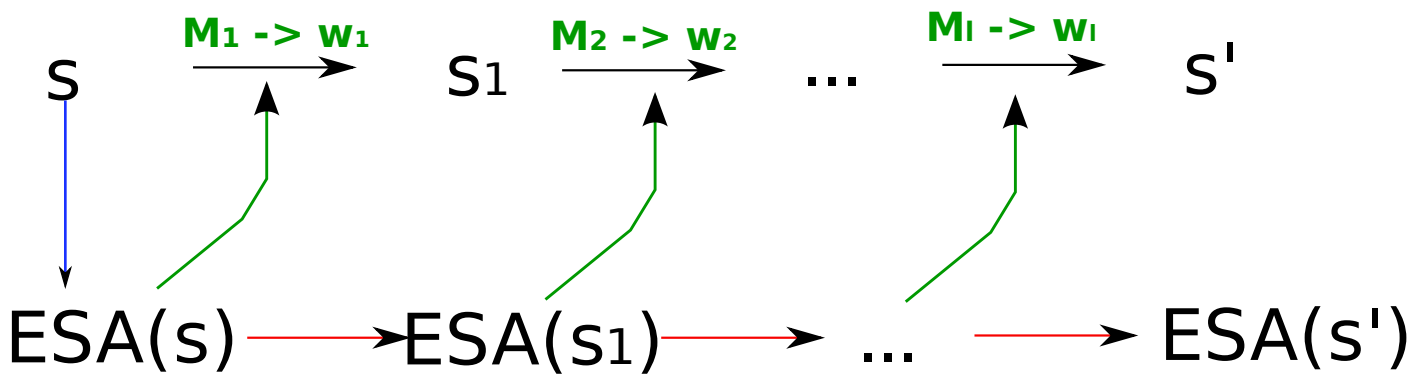
GI algorithm using ESA

Replace some (all) occurrences of the same repeat by a new character at each step



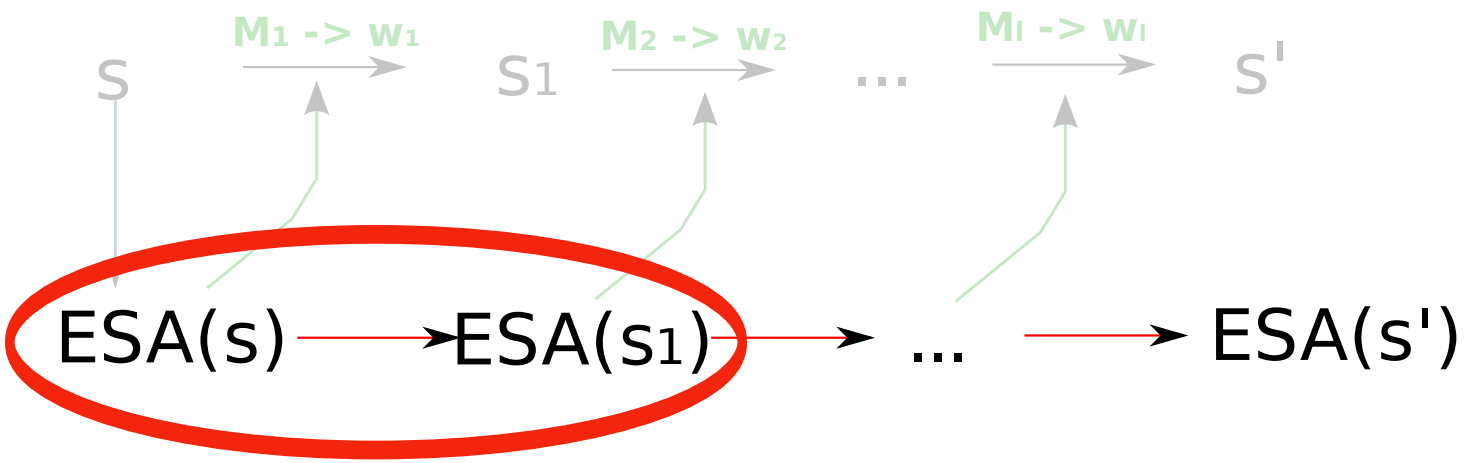
GI algorithm using ESA

Replace some (all) occurrences of the same repeat by a new character at each step



GI algorithm using ESA

Replace some (all) occurrences of the same repeat by a new character at each step



Update of Indexing Structures

- Lots of work done on Dynamic Dictionaries
  - Sahinalp & Vishkin “Efficient Approximate and Dynamic Matching of Patterns Using a Labeling Paradigm” 1996
  - Ferragini & Grossi “Fast incremental text editing” 1996
- Apostolico & Lonardi couldn’t «find an existing satisfactory solution to modifying our»Minimal Augmented Suffix Tree.
  - “Some theory and practice of greedy off-line textual substitution” 1998
- Lanctot, Li & Yang (GTAC) do it for suffix tree . . . , but just for the longest repeat
  - “Estimating DNA Sequence Entropy” 2000
- Nevill-Manning & Witten perform a bubble sort to update a suffix array
  - “On-line and off-line heuristics for inferring hierarchies of repetitions in sequences” 2000

Our update algorithm

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGCATATCATC
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	3	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATATCATC
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCATATCATC
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCATATCATC
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	2	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

- Enhanced Suffix array for *ACGCATCTCCATCGCGCATATCATC*

Our update algorithm

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGCATATCATC
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	3	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATATCATC
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCATATCATC
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCATATCATC
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	2	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

- Enhanced Suffix array for *ACGCATCTCCATCGCGCATATCATC*

- Replace each occurrence of $w = CAT$ by M.

Our update algorithm

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGCATATCATC
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	3	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATATCATC
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCATATCATC
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCATATCATC
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	2	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

Steps of the algorithm

1. Delete positions
2. Move some lines
3. Update LCP

1. Delete some lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	0	25		
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	3	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	2	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

- Delete lines *inside* the repetitions

1. Delete some lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	0	25		
1	14	0	0	ACGCATCTCCATCGCGC…
~~2~~	~~18~~	~~1~~	~~17~~	~~ATATCATC~~
3	11	2	22	ATC
4	6	3	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	2	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

- Delete lines *inside* the repetitions
- For all *intern position j*:
  - Delete line $i = isa[j]$:

1. Delete some lines

ACGCATCTCCATCGCGCATATCATC

i	next	prev	isa	lcp	sa	suffix
0	1		0	25		
1	3	0	14	0	0	ACGCATCTCCATCGCGC...
2	0	1	18	1	17	ATATCATC
3	4	1	11	2	22	ATC
4	5	3	6	3	19	ATCATC
5	6	4	25	3	10	ATCGCGCATATCATC
6	7	5	16	3	4	ATCTCCATCGCGCATAT...
7	8	6	23	0	24	C
8	9	7	12	1	16	CATATCATC
9	10	8	10	3	21	CATC
10	11	9	5	4	9	CATCGCGCATATCATC
11	12	10	24	4	3	CATCTCCATCGCGCAT...
12	13	11	15	1	8	CCATCGCGCATATCATC
13	14	12	19	1	14	CGCATATCATC
14	15	13	13	5	1	CGCATCTCCATCGCGCA...
15	16	14	17	3	12	CGCGCATATCATC
16	17	15	8	1	6	CTCCATCGCGCATATCATC
17	18	16	2	0	15	GCATATCATC
18	19	17	20	4	2	GCATCTCCATCGCGCAT...
19	20	18	4	2	13	GCGCATATCATC
20	21	19	22	0	18	TATCATC
21	22	20	9	1	23	TC
22	23	21	3	2	20	TCATC
23	24	22	21	2	7	TCCATCGCGCATATCATC
24	25	23	7	2	11	TCGCGCATATCATC
25		24	0	2	5	TCTCCATCGCGCATATCATC

- Delete lines *inside* the repetitions
- For all *intern position j*:
  - Delete line $i = isa[j]$:
    $next[prev[i]] \leftarrow next[i]$
    $prev[next[i]] \leftarrow prev[i]$

1. Delete some lines

ACGCATCTCCATCGCGCATATCATC

i	next	prev	isa	lcp	sa	suffix
0	1		0	25		
1	~~3~~	0	14	0	0	ACGCATCTCCATCGCGC…
~~2~~	~~0~~	~~1~~	~~18~~	~~1~~	~~17~~	~~ATATCATC~~
3	4	~~1~~	11	2	22	ATC
4	5	3	6	3	19	ATCATC
5	6	4	25	3	10	ATCGCGCATATCATC
6	7	5	16	3	4	ATCTCCATCGCGCATAT…
7	8	6	23	0	24	C
8	9	7	12	1	16	CATATCATC
9	10	8	10	3	21	CATC
10	11	9	5	4	9	CATCGCGCATATCATC
11	12	10	24	4	3	CATCTCCATCGCGCAT…
12	13	11	15	1	8	CCATCGCGCATATCATC
13	14	12	19	1	14	CGCATATCATC
14	15	13	13	5	1	CGCATCTCCATCGCGCA…
15	16	14	17	3	12	CGCGCATATCATC
16	17	15	8	1	6	CTCCATCGCGCATATCATC
17	18	16	2	0	15	GCATATCATC
18	19	17	20	4	2	GCATCTCCATCGCGCAT…
19	20	18	4	2	13	GCGCATATCATC
20	21	19	22	0	18	TATCATC
21	22	20	9	1	23	TC
22	23	21	3	2	20	TCATC
23	24	22	21	2	7	TCCATCGCGCATATCATC
24	25	23	7	2	11	TCGCGCATATCATC
25		24	0	2	5	TCTCCATCGCGCATATCATC

- Delete lines *inside* the repetitions
- Update LCP
- For all *intern position j*:
  - Delete line $i = isa[j]$:
    $next[prev[i]] \leftarrow next[i]$
    $prev[next[i]] \leftarrow prev[i]$

1. Delete some lines

ACGCATCTCCATCGCGCATATCATC

i	next	prev	isa	lcp	sa	suffix
0	1		0	25		
1	3	0	14	0	0	ACGCATCTCCATCGCGC...
~~2~~	~~0~~	~~1~~	~~13~~	~~1~~	~~17~~	~~ATATCATC~~
3	4	1	11	1	22	ATC
4	5	3	6	3	19	ATCATC
5	6	4	25	3	10	ATCGCGCATATCATC
6	7	5	16	3	4	ATCTCCATCGCGCATAT...
7	8	6	23	0	24	C
8	9	7	12	1	16	CATATCATC
9	10	8	10	3	21	CATC
10	11	9	5	4	9	CATCGCGCATATCATC
11	12	10	24	4	3	CATCTCCATCGCGCAT...
12	13	11	15	1	8	CCATCGCGCATATCATC
13	14	12	19	1	14	CGCATATCATC
14	15	13	13	5	1	CGCATCTCCATCGCGCA...
15	16	14	17	3	12	CGCGCATATCATC
16	17	15	8	1	6	CTCCATCGCGCATATCATC
17	18	16	2	0	15	GCATATCATC
18	19	17	20	4	2	GCATCTCCATCGCGCAT...
19	20	18	4	2	13	GCGCATATCATC
20	21	19	22	0	18	TATCATC
21	22	20	9	1	23	TC
22	23	21	3	2	20	TCATC
23	24	22	21	2	7	TCCATCGCGCATATCATC
24	25	23	7	2	11	TCGCGCATATCATC
25		24	0	2	5	TCTCCATCGCGCATATCATC

- Delete lines *inside* the repetitions
- Update LCP
- For all *intern position j*:
  - Delete line $i = isa[j]$:
    $next[prev[i]] \leftarrow next[i]$
    $prev[next[i]] \leftarrow prev[i]$
  - $LCP[next[i]] \leftarrow min(LCP[i], LCP[next[i]])$

1. Delete some lines

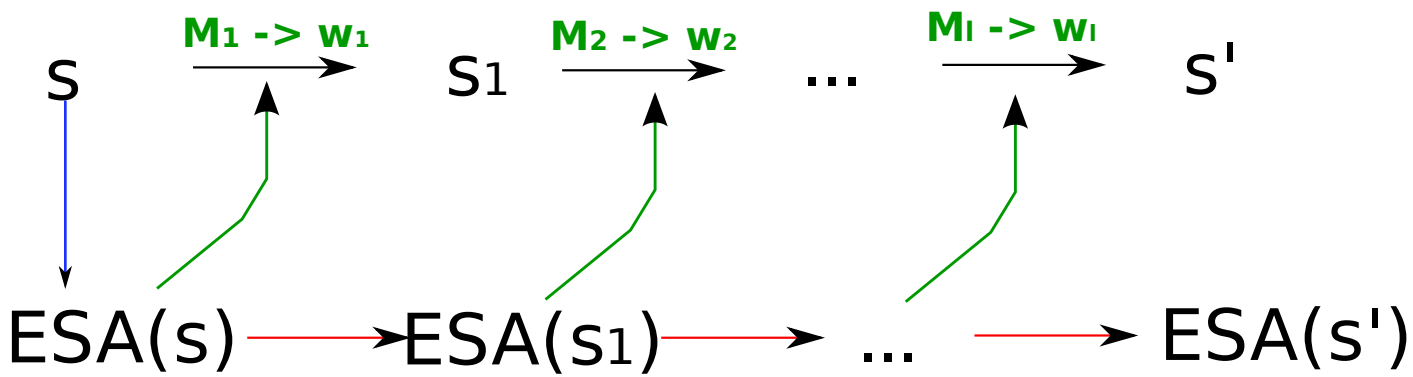
ACGCATCTCCATCGCGCATATCATC

i	next	prev	isa	lcp	sa	suffix
0	1		0	25		
1	4	0	14	0	0	ACGCATCTCCATCGCGC…
~~2~~	~~3~~	~~1~~	~~18~~	~~1~~	~~17~~	~~ATATCATC~~
~~3~~	~~1~~	~~11~~	~~11~~	~~1~~	~~22~~	~~ATC~~
4	7	0	6	1	19	ATCATC
~~5~~	~~6~~	~~1~~	~~25~~	~~8~~	~~10~~	~~ATCGCGCATATCATC~~
~~6~~	~~7~~	~~5~~	~~10~~	~~8~~	~~1~~	~~ATCTCCATCGCGCATAT…~~
7	8	4	23	0	24	C
8	9	7	12	1	16	CATATCATC
9	10	8	10	3	21	CATC
10	11	9	5	4	9	CATCGCGCATATCATC
11	12	10	24	4	3	CATCTCCATCGCGCAT…
12	13	11	15	1	8	CCATCGCGCATATCATC
13	14	12	19	1	14	CGCATATCATC
14	15	13	13	5	1	CGCATCTCCATCGCGCA…
15	16	14	17	3	12	CGCGCATATCATC
16	17	15	8	1	6	CTCCATCGCGCATATCATC
17	18	16	2	0	15	GCATATCATC
18	19	17	20	4	2	GCATCTCCATCGCGCAT…
19	22	18	4	2	13	GCGCATATCATC
~~20~~	~~21~~	~~19~~	~~22~~	~~0~~	~~18~~	~~TATCATC~~
~~21~~	~~22~~	~~20~~	~~0~~	~~1~~	~~23~~	~~TC~~
22	23	19	3	0	20	TCATC
23	24	22	21	2	7	TCCATCGCGCATATCATC
~~24~~	~~25~~	~~23~~	~~7~~	~~2~~	~~11~~	~~TCGCGCATATCATC~~
~~25~~		~~24~~	~~0~~	~~2~~	~~5~~	~~TCTCCATCGCGCATATCATC~~

- Delete lines *inside* the repetitions
- Update LCP
- For all *intern position j*:
  - Delete line $i = isa[j]$:
    $next[prev[i]] \leftarrow next[i]$
    $prev[next[i]] \leftarrow prev[i]$
  - $LCP[next[i]] \leftarrow min(LCP[i], LCP[next[i]])$

GI algorithm using ESA

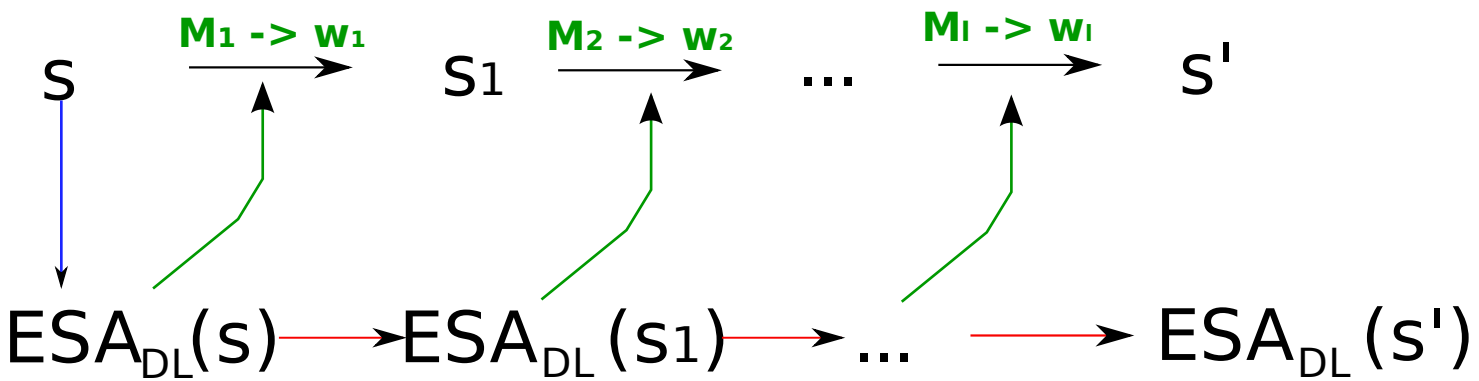
Replace some (all) occurrences of the same repeat by a new character at each step



GI algorithm using ESA

Replace some (all) occurrences of the same repeat by a new character at each step

$$ESA_{DL} = ESA + next + prev$$



2. Move lines

- Introduction of new symbol (M)
- with new lexicographical order (we choosed $M > a \, \forall a \in \Sigma$)
- enhanced suffix array is disordered
- must move some lines

2. Move lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

2. Move lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

CAT [8,11]

2. Move lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

CAT [8,11]

2. Move lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	1	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

CAT [8,11]

2. Move lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC

CAT [8,11]

2. Move lines

ACGCATCTCCATCGCGCATATCATC

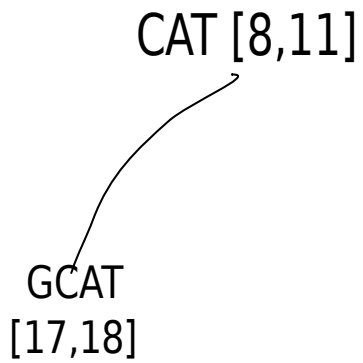
i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

CAT [8,11]

2. Move lines

ACGCATCTCCATCGCGCATATCATC

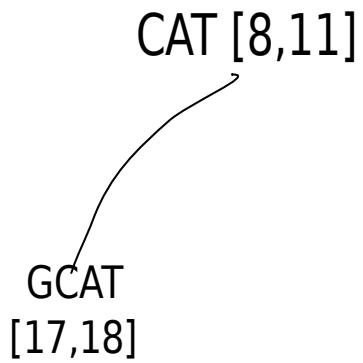
i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…



2. Move lines

ACGCATCTCCATCGCGCATATCATC

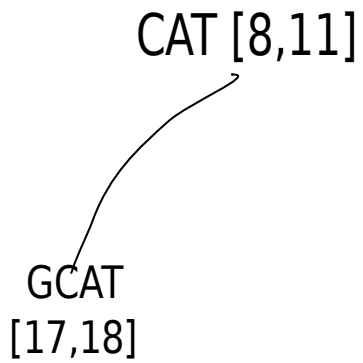
i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	0	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	2	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…



2. Move lines

ACGCATCTCCATCGCGCATATCATC

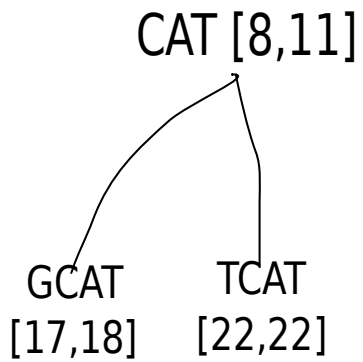
i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	0	20	TCATC
23	21	2	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…



2. Move lines

ACGCATCTCCATCGCGCATATCATC

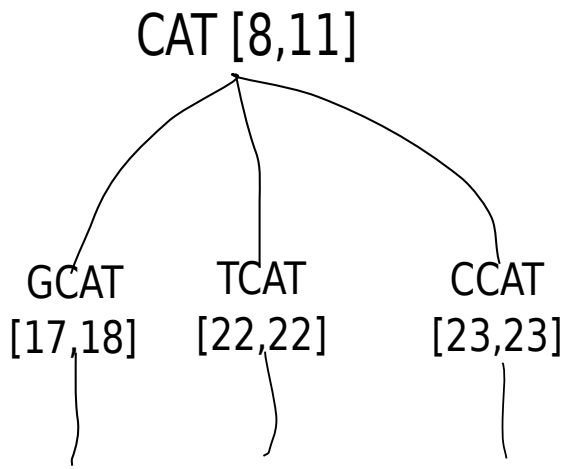
i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	1	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…



2. Move lines

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	3	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

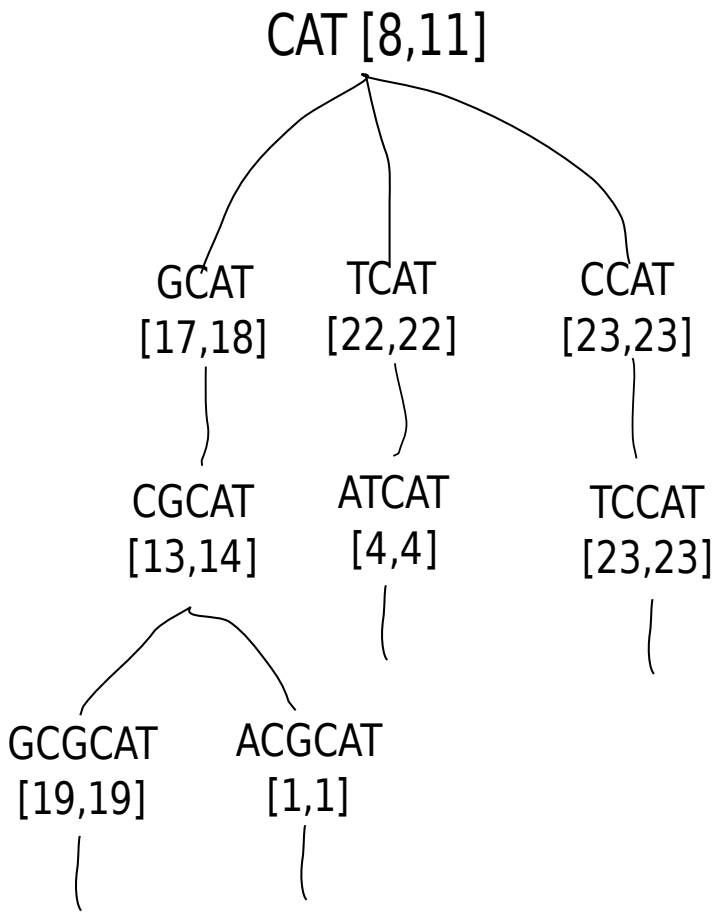


2. Move lines

ACGCATCTCCATCGCGCATATCATC

Left Context Tree(w) of sequence s (LCT)

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	1	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

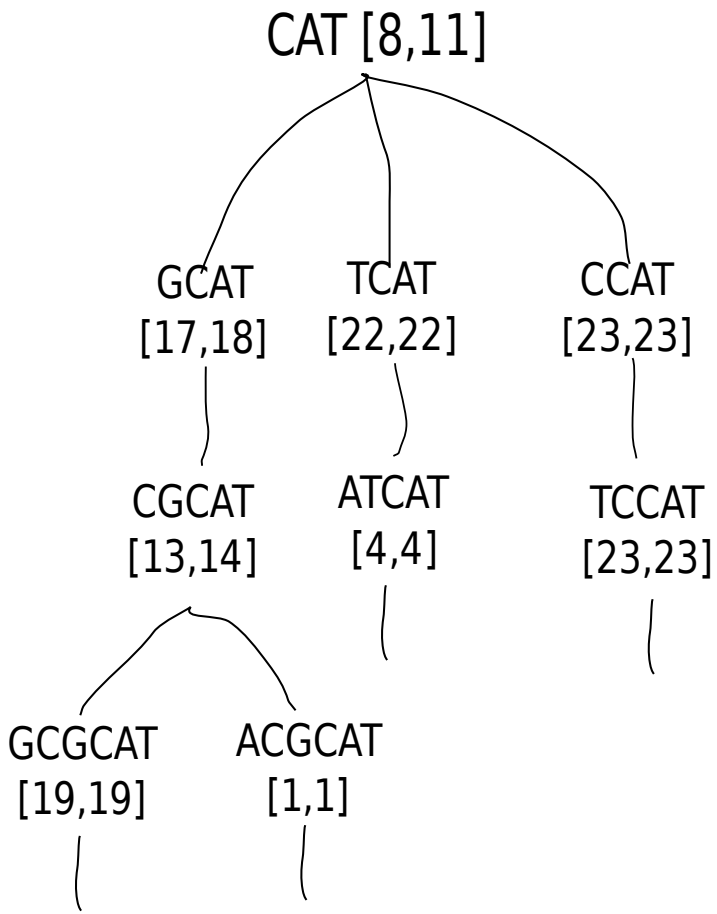


2. Move lines

ACGCATCTCCATCGCGCATATCATC

Left Context Tree(w) of sequence s (LCT)

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	1	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

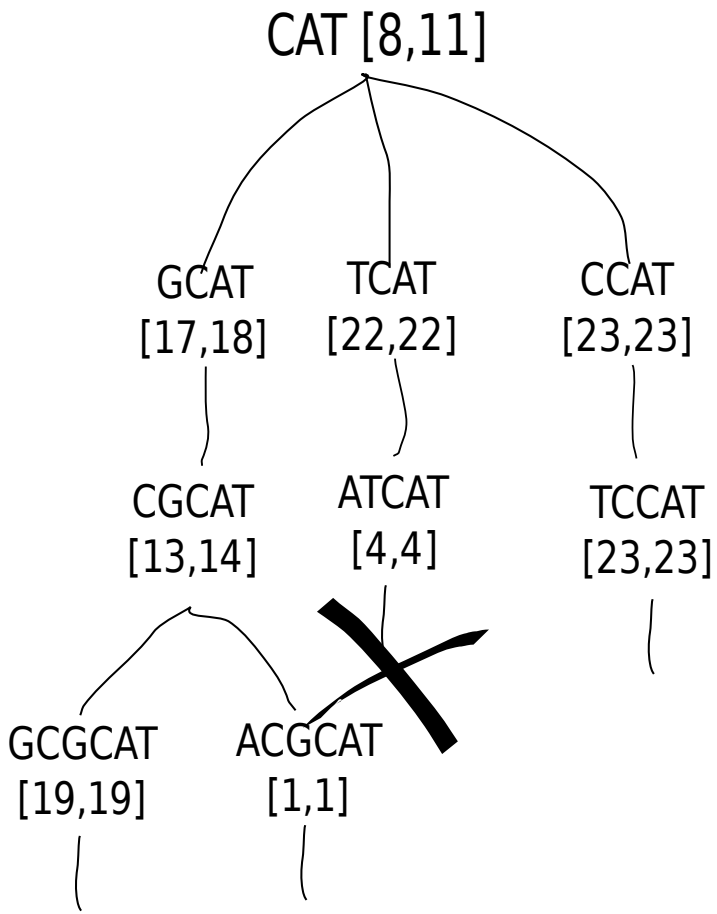


2. Move lines

ACGCATCTCCATCGCGCATATCATC

Left Context Tree(w) of sequence s (LCT)

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	1	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

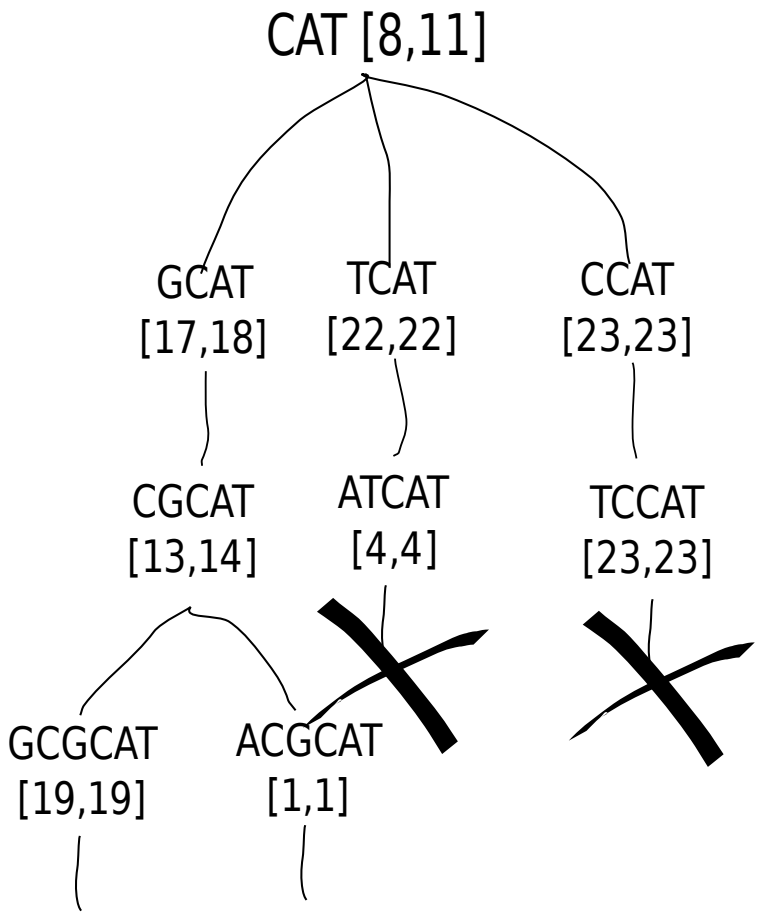


2. Move lines

ACGCATCTCCATCGCGCATATCATC

Left Context Tree(w) of sequence s (LCT)

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	1	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

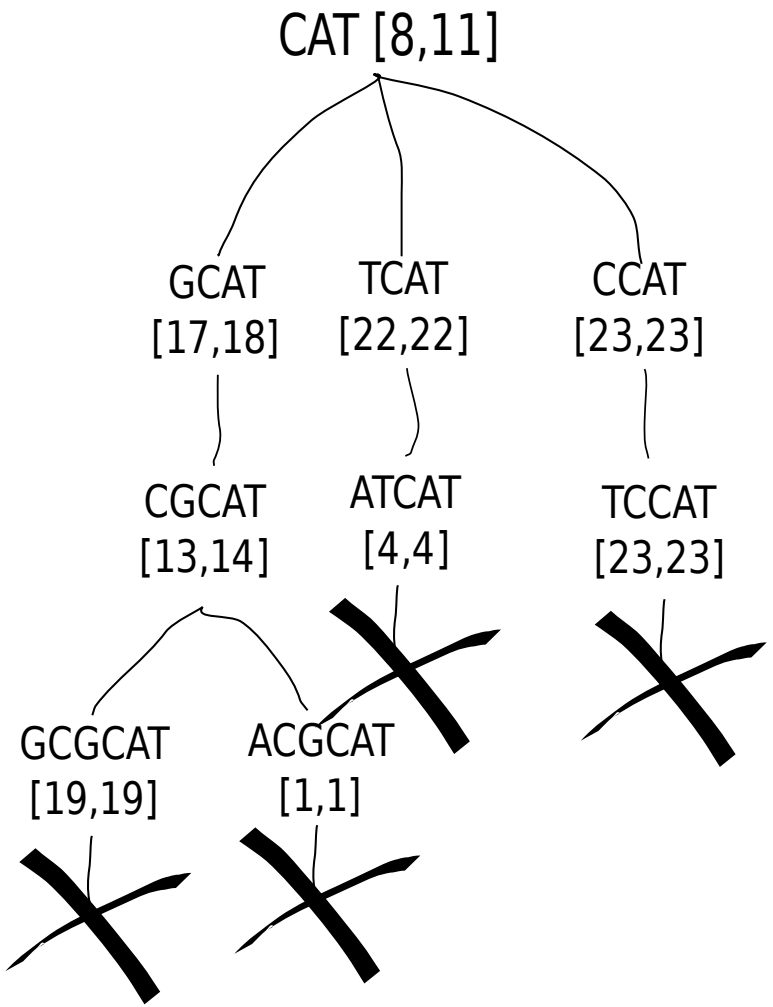


2. Move lines

ACGCATCTCCATCGCGCATATCATC

Left Context Tree(*w*) of sequence *s* (LCT)

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
12	15	1	8	CCATCGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
15	17	1	12	CGCGCATATCATC
16	8	1	6	CTCCATCGCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
19	4	0	13	GCGCATATCATC
20	22	0	18	TATCATC
21	9	1	23	TC
22	3	1	20	TCATC
23	21	0	7	TCCATCGCGCATATCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…



3. Update LCP

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
15	17	1	12	CGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
16	8	1	6	CTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
19	4	0	13	GCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
20	22	0	18	TATCATC
21	9	1	23	TC
23	21	0	7	TCCATCGCGCATATCATC
22	3	1	20	TCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

A repetition colapse in one character:

$|\omega| \longrightarrow 1$

3. Update LCP

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
15	17	1	12	CGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
16	8	1	6	CTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
19	4	0	13	GCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
20	22	0	18	TATCATC
21	9	1	23	TC
23	21	0	7	TCCATCGCGCATATCATC
22	3	1	20	TCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

Inside a group:

3. Update LCP

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
15	17	1	12	CGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
16	8	1	6	CTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
19	4	0	13	GCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
20	22	0	18	TATCATC
21	9	1	23	TC
23	21	0	7	TCCATCGCGCATATCATC
22	3	1	20	TCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	3	21	CATC
10	5	4	9	CATCGCGCATATCATC
11	24	4	3	CATCTCCATCGCGCAT…

Inside a group:

- For group of the root-interval: recalculate

3. Update LCP

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
15	17	1	12	CGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
16	8	1	6	CTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
19	4	0	13	GCGCATATCATC
17	2	1	15	GCATATCATC
18	20	4	2	GCATCTCCATCGCGCAT…
20	22	0	18	TATCATC
21	9	1	23	TC
23	21	0	7	TCCATCGCGCATATCATC
22	3	1	20	TCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	1	21	CATC
10	5	2	9	CATCGCGCATATCATC
11	24	2	3	CATCTCCATCGCGCAT…

Inside a group:

- For group of the root-interval: recalculate

3. Update LCP

ACGCATCTCCATCGCGCATATCATC

i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
15	17	1	12	CGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	5	1	CGCATCTCCATCGCGCA…
16	8	1	6	CTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
19	4	0	13	GCGCATATCATC
17	2	1	15	GCATATCATC
18	20	2	2	GCATCTCCATCGCGCAT…
20	22	0	18	TATCATC
21	9	1	23	TC
23	21	0	7	TCCATCGCGCATATCATC
22	3	1	20	TCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	1	21	CATC
10	5	2	9	CATCGCGCATATCATC
11	24	2	3	CATCTCCATCGCGCAT…

Inside a group:

- For group of the root-interval: recalculate
- For other groups: use information of root group

3. Update LCP

ACGCATCTCCATCGCGCATATCATC

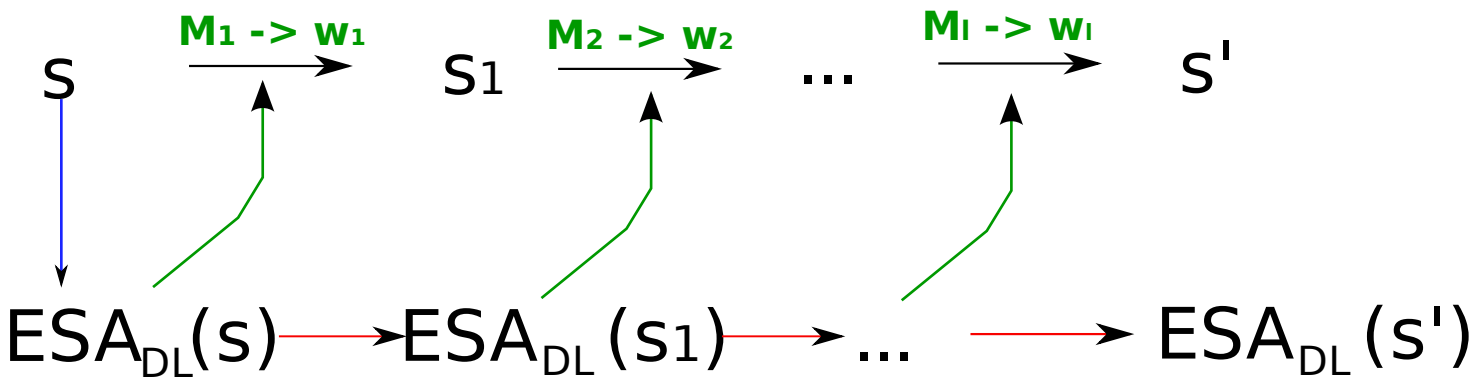
i	isa	lcp	sa	suffix
0	1	0	25	
1	14	0	0	ACGCATCTCCATCGCGC…
2	18	1	17	ATATCATC
3	11	2	22	ATC
4	6	1	19	ATCATC
5	25	3	10	ATCGCGCATATCATC
6	16	3	4	ATCTCCATCGCGCATAT…
7	23	0	24	C
15	17	1	12	CGCGCATATCATC
13	19	2	14	CGCATATCATC
14	13	3	1	CGCATCTCCATCGCGCA…
16	8	1	6	CTCCATCGCGCATATCATC
12	15	1	8	CCATCGCGCATATCATC
19	4	0	13	GCGCATATCATC
17	2	1	15	GCATATCATC
18	20	2	2	GCATCTCCATCGCGCAT…
20	22	0	18	TATCATC
21	9	1	23	TC
23	21	0	7	TCCATCGCGCATATCATC
22	3	1	20	TCATC
24	7	2	11	TCGCGCATATCATC
25	0	2	5	TCTCCATCGCGCATATCATC
8	12	0	16	CATATCATC
9	10	1	21	CATC
10	5	2	9	CATCGCGCATATCATC
11	24	2	3	CATCTCCATCGCGCAT…

Inside a group:

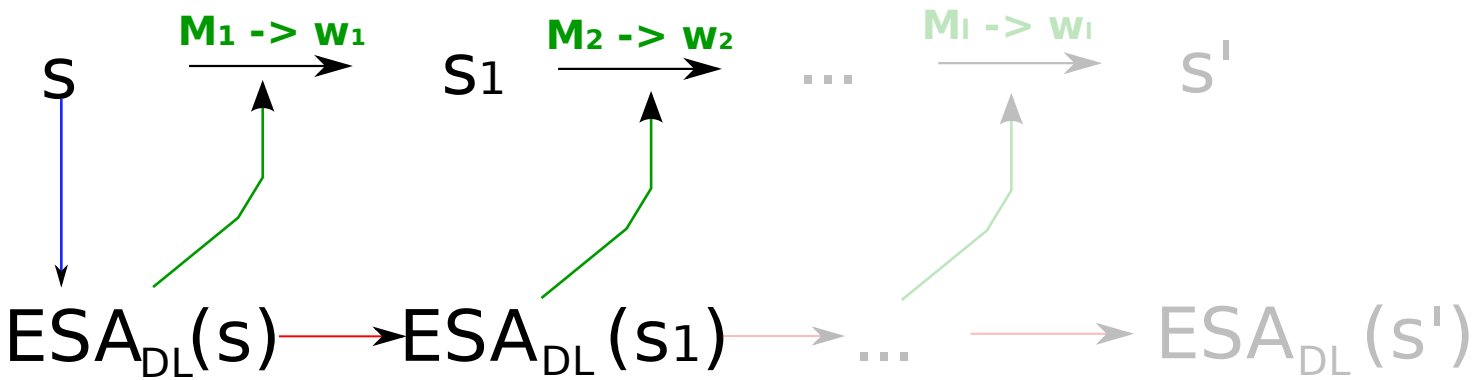
- For group of the root-interval: recalculate
- For other groups: use information of root group

Efficiency

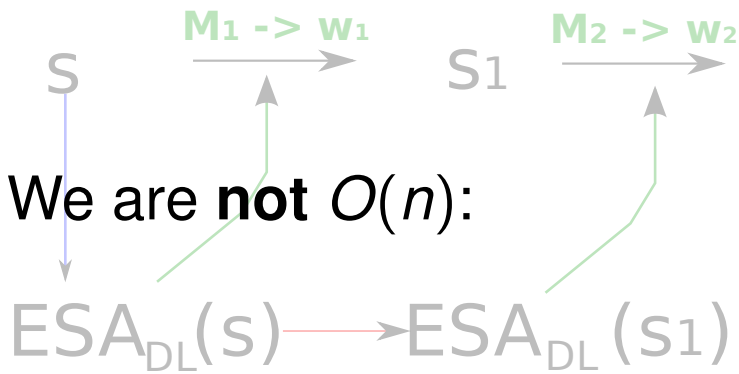
Efficiency



Efficiency

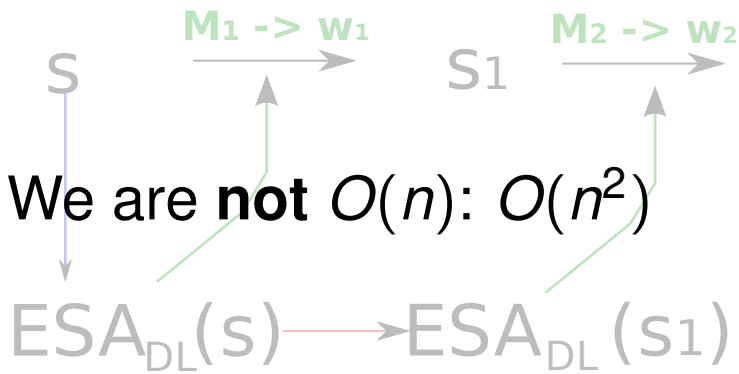


Efficiency



- We are **not** $O(n)$:

Efficiency



- We are **not** $O(n)$: $O(n^2)$

Efficiency

- We are **not** $O(n)$: $O(n^2)$
- Suffix Array creation algorithm

Efficiency

- We are **not** $O(n)$: $O(n^2)$

- Suffix Array creation algorithm

Kärkkäinen and Sanders[a]	2003	$O(n)$	*K & S*
Larsson and Sadakane[b]	1999	$O(log(n)n)$	*L & S*

[a] J. Kärkkäinen and P. Sanders: Simple linear work suffix array construction, in Proc. ICALP, Springer, 2003

[b] N. J. Larsson and K. Sadakane: Faster suffix sorting, Tech. Rep. LU-CS-TR:99-214, LUNDFD6/(NFCS-3140)/1-20/(1999), Department of Computer Science, Lund University, Sweden, May 1999

Efficiency

Experiments

Efficiency

Experiments

- Different corpora used in data compression:
  - calgary [Bell, Cleary & Witten “Text Compression” 1990]
  - canterbury [Arnold & Bell “ A corpus for the evaluation of lossless compression algorithm” 1997]
  - large canterbury
  - manzini’s historical [Mazini & Rastero “A Simple and Fast DNA Compressor” 2004]
  - purdue [Apostolico & Lonardi “Compression of biological sequences by greedy off-line textual substitution” 2000]

Efficiency

Experiments

- Different corpora used in data compression:
  - calgary [Bell, Cleary & Witten “Text Compression” 1990]
  - canterbury [Arnold & Bell “ A corpus for the evaluation of lossless compression algorithm” 1997]
  - large canterbury
  - manzini’s historical [Mazini & Rastero “A Simple and Fast DNA Compressor” 2004]
  - purdue [Apostolico & Lonardi “Compression of biological sequences by greedy off-line textual substitution” 2000]
- Different ways of choosing the repetitions:
  - random
  - maximal length
  - maximal compression

Efficiency

- 500 steps of GI algorithm
- measured (user + sys) time
- ratio $\frac{\text{recreating from scratch}}{\text{our update}}$

Good news

$$\frac{\text{recreating from scratch}}{\text{our update}} = \begin{array}{c} \uparrow \\ 1.0 \\ \downarrow \end{array} \begin{array}{l} \textit{better} \\ \\ \textit{worse} \end{array}$$

sequence	size	Φ lcp	random		max length		max comp.	
			K&S	L&S	K&S	L&S	K&S	L&S
bible.txt	4MB	13,0						
E.coli	4.6MB	23,0						
world192	2.5MB	17,4						

Good news

$$\frac{\text{recreating from scratch}}{\text{our update}} = \begin{array}{c} \uparrow \\ 1.0 \\ \downarrow \end{array} \begin{array}{l} \textit{better} \\ \\ \textit{worse} \end{array}$$

sequence	size	Φ lcp	random		max length		max comp.	
			K&S	L&S	K&S	L&S	K&S	L&S
bible.txt	4MB	13,0	**66,8**	**22,9**	**64,4**	**22,5**	**15,4**	**3,7**
E.coli	4.6MB	23,0	**69,1**	**27,4**	**53,5**	**24,0**	**9,5**	**2,1**
world192	2.5MB	17,4	**65,0**	**21,8**	**60,7**	**21,1**	**16,3**	**4,5**

Good news

$$\frac{\text{recreating from scratch}}{\text{our update}} = \begin{array}{c} \uparrow \\ 1.0 \\ \downarrow \end{array} \begin{array}{l} \textit{better} \\ \\ \textit{worse} \end{array}$$

sequence	size	Φ lcp	random		max length		max comp.	
			K&S	L&S	K&S	L&S	K&S	L&S
bible.txt	4MB	13,0	**66,8**	**22,9**	**64,4**	**22,5**	**15,4**	**3,7**
E.coli	4.6MB	23,0	**69,1**	**27,4**	**53,5**	**24,0**	**9,5**	**2,1**
world192	2.5MB	17,4	**65,0**	**21,8**	**60,7**	**21,1**	**16,3**	**4,5**

Good news

$$\frac{\text{recreating from scratch}}{\text{our update}} = \begin{array}{l} \uparrow \quad \textit{better} \\ 1.0 \\ \downarrow \quad \textit{worse} \end{array}$$

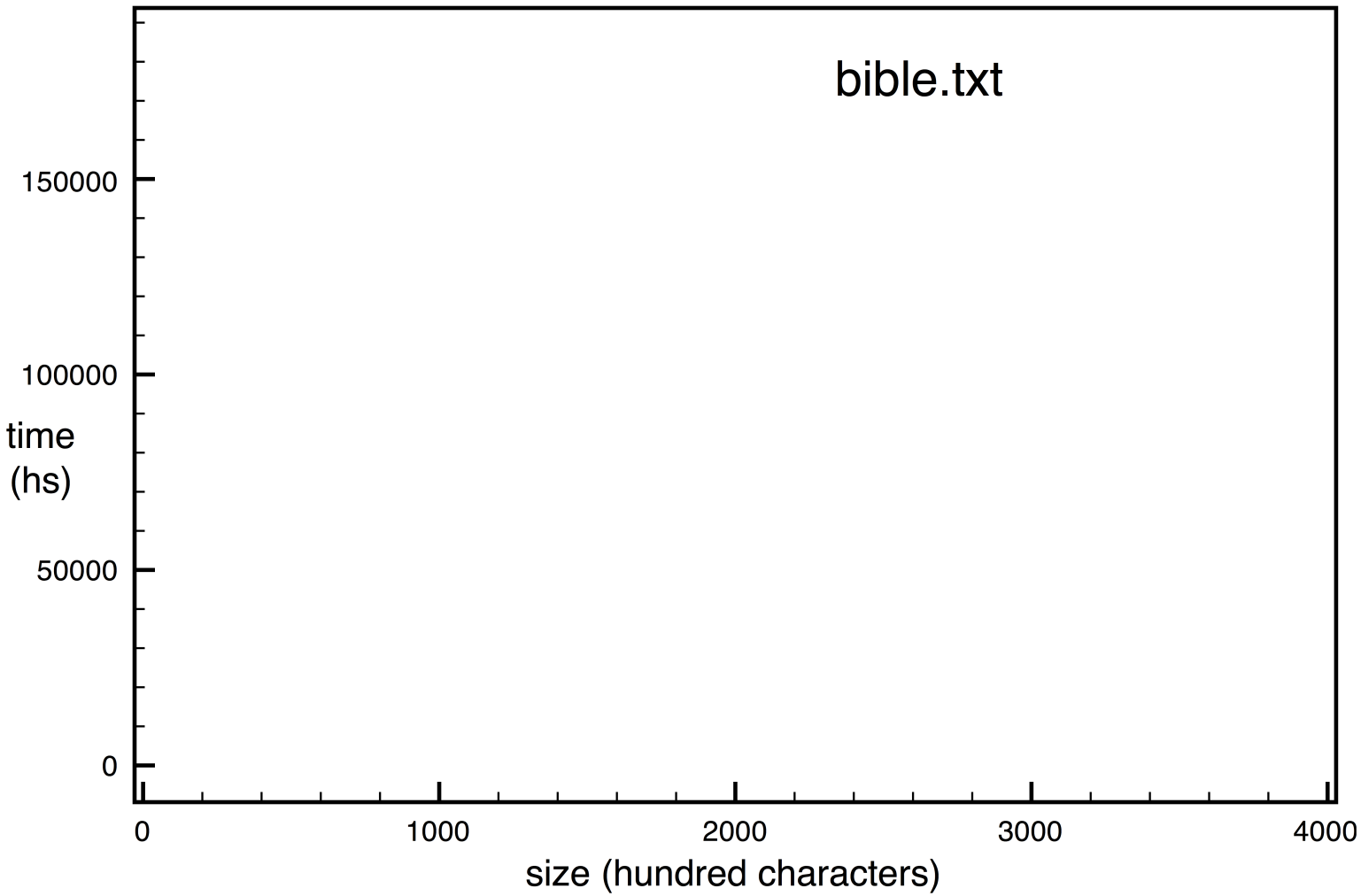
sequence	size	Φ lcp	random		max length		max comp.	
			K&S	L&S	K&S	L&S	K&S	L&S
bible.txt	4MB	13,0	**66,8**	**22,9**	**64,4**	**22,5**	**15,4**	**3,7**
E.coli	4.6MB	23,0	**69,1**	**27,4**	**53,5**	**24,0**	**9,5**	**2,1**
world192	2.5MB	17,4	**65,0**	**21,8**	**60,7**	**21,1**	**16,3**	**4,5**

Two Bad News

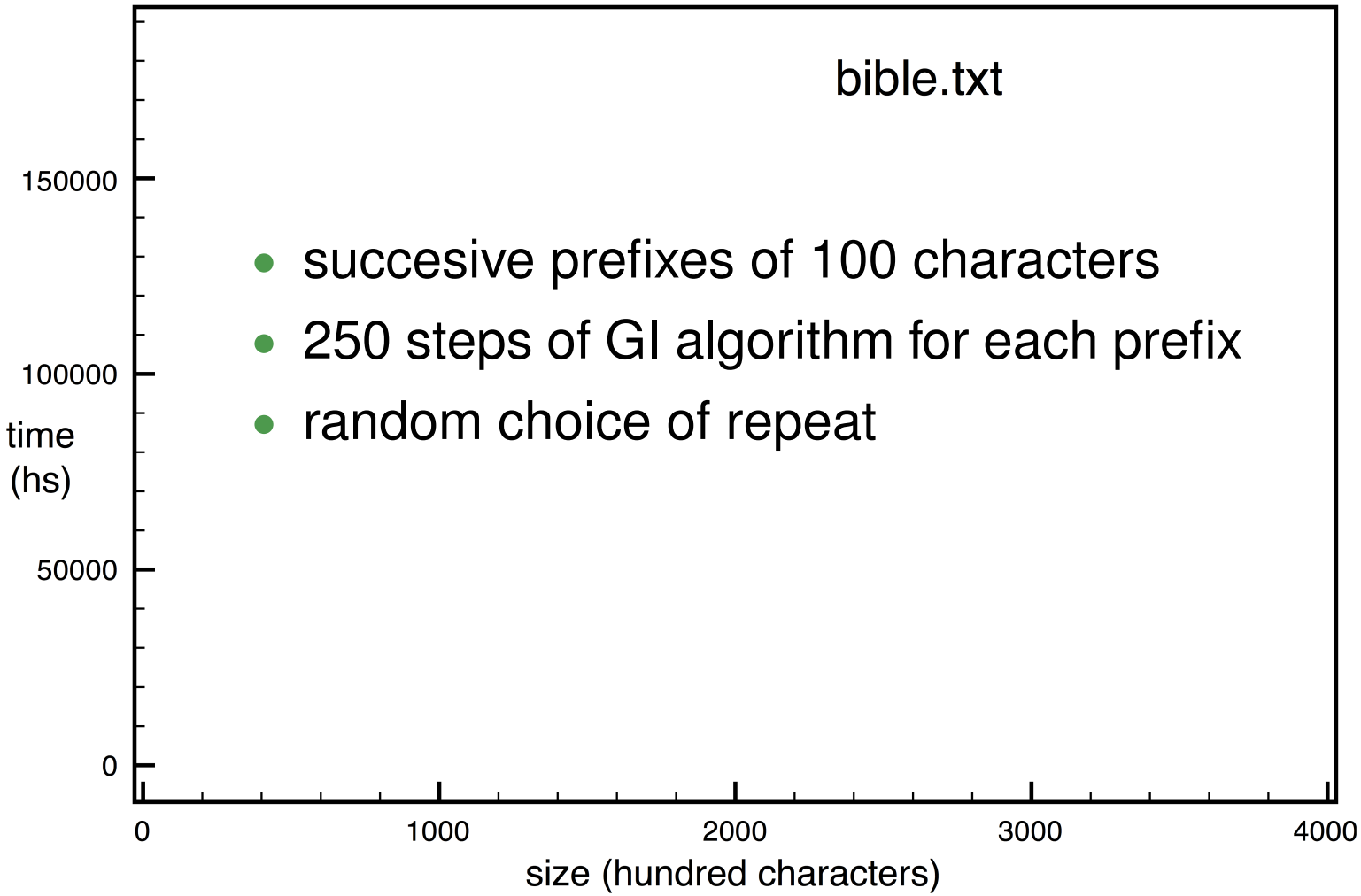
$$\frac{\text{recreating from scratch}}{\text{our update}} = \begin{array}{l} \uparrow \quad \textit{better} \\ 1.0 \\ \downarrow \quad \textit{worse} \end{array}$$

sequence	size	Φ lcp	random		max length		max comp	
			K&S	L&S	K&S	L&S	K&S	L&S
ptt5	513216	2353	**6,0**	**3,7**	**7,7**	**5,3**	**0,4**	**0,2**
Spor_All_2x	444906	56022	**0,7**	**0,8**	**0,6**	**0,8**	**0,6**	**1,2**

Practical Complexity

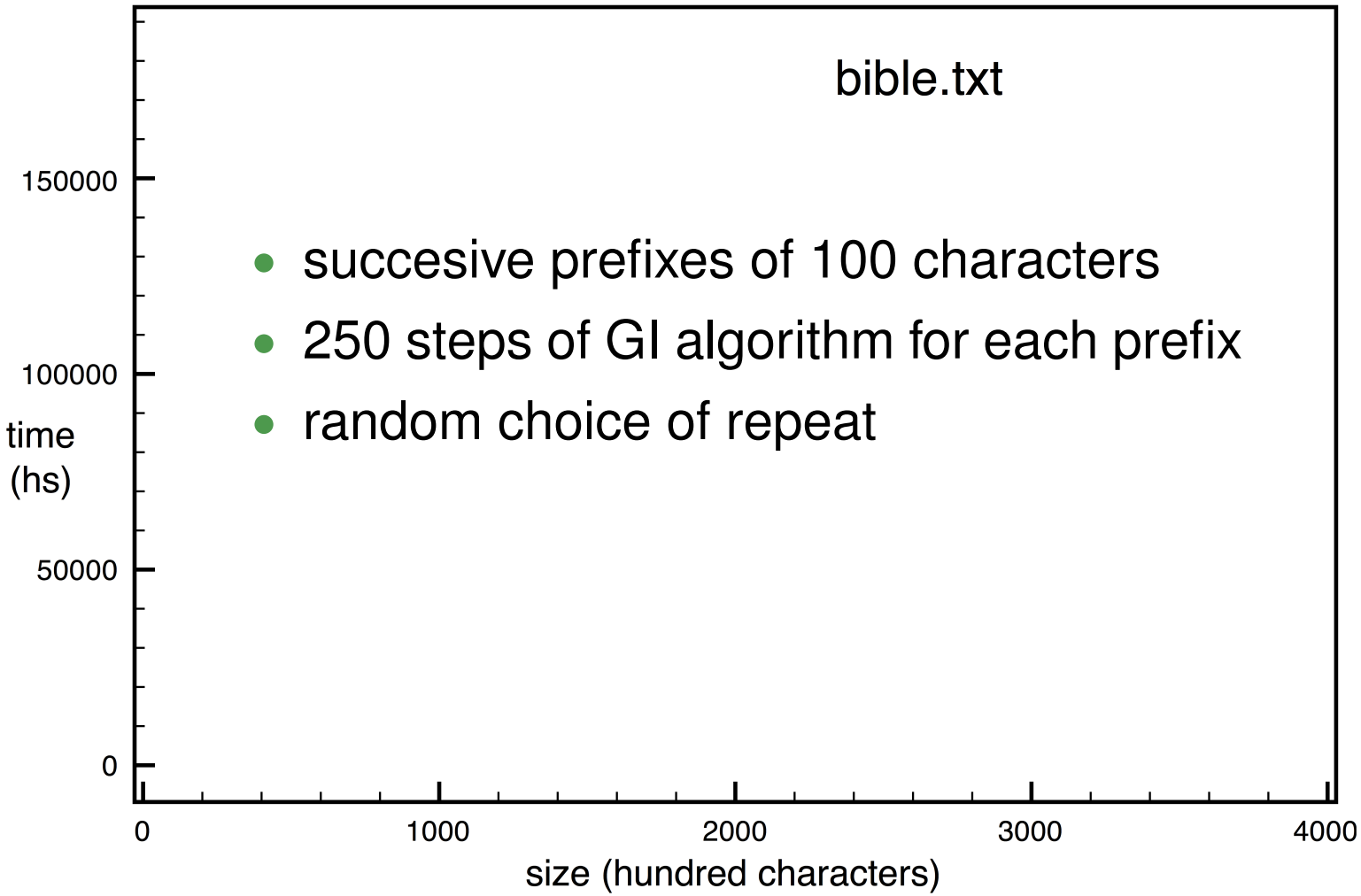


Practical Complexity



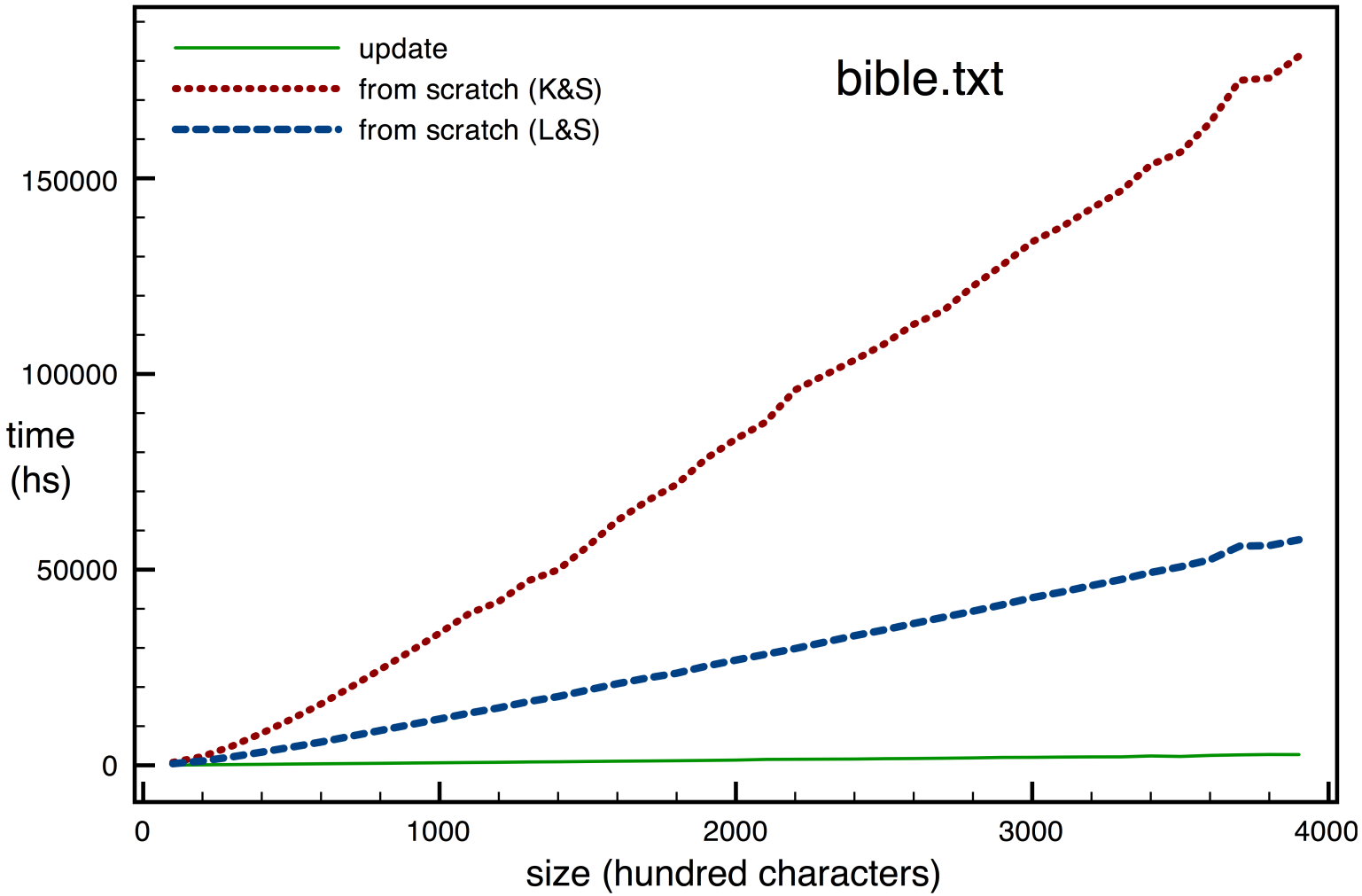
- succesive prefixes of 100 characters
- 250 steps of GI algorithm for each prefix
- random choice of repeat

Practical Complexity



- succesive prefixes of 100 characters
- 250 steps of GI algorithm for each prefix
- random choice of repeat

Practical Complexity



Conclusion

- Purpose: replace iteratively a list of occurrences of a repeat by a new character
- Developed an algorithm for updating a ESA_{DL}
- “Independent” from size of sequence
- Dependent from:
  - repetition $\rightarrow$ coverage
  - sequence $\rightarrow$ lcp

Conclusion

- Purpose: replace iteratively a list of occurrences of a repeat by a new character
- Developed an algorithm for updating a ESA_{DL}
- “Independent” from size of sequence
- Dependent from:
  - repetition $\rightarrow$ coverage
  - sequence $\rightarrow$ lcp

Perspectives

- When use update algorithm / recreation from scratch ?
- parameter $\rightarrow$ hybrid algorithm
- Complexity of whole process

To know more

- `http://www.irisa.fr/symbiose/mgalle/suffix_array_update`
- Ask Questions

Random

sequence	size (chars.)	update	K & S	L & S	ratio K & S	ratio L & S
CANTERBURY CORPUS						
alice29.txt	152089	163	2812	1497	17.25	9.18
asyoulik.txt	125179	131	2111	1109	16.11	8.47
cp.html	24603	15	132	95	8.80	6.33
fields.c	11150	6	38	31	6.33	5.17
grammar.lsp	3721	3	5	5	1.67	1.67
kennedy.xls	1029744	1323	34905	12829	26.38	9.70
lcet10.txt	426754	516	17151	6871	33.24	13.32
plrabn12.txt	481861	588	22657	8853	38.53	15.06
ptt5	513216	1248	7389	4617	5.92	3.70
sum	38240	42	234	151	5.57	3.60
xargs.1	4227	6	25	9	4.17	1.50
LARGE CORPUS						
bible.txt	4047392	5055	337725	115481	66.81	22.84
E.coli	4638690	5534	382636	151405	69.14	27.36
world192.txt	2473400	3084	200643	67079	65.06	21.75
PURDUE CORPUS						
All_Up_1M.fasta	1001002	1238	61657	24597	49.80	19.87
All_Up_400k.fasta	399615	501	13959	6777	27.86	13.53
Helden_All.fasta	112507	119	1511	963	12.70	8.09
Helden_CGN.fasta	32871	31	244	172	7.87	5.55
Spor_All_2x.fasta	444906	112	82	94	0.73	0.84
Spor_All.fasta	222453	246	3658	2107	14.87	8.57
Spor_EarlyI.fasta	31039	34	187	152	5.50	4.47
Spor_EarlyII.fasta	25008	20	145	151	7.25	7.55
Spor_Middle.fasta	54325	S 51	526	351	10.31	6.88

Maximal Length

sequence	size (chars.)	update	K & S	L & S	ratio K & S	ratio L & S
CANTERBURY CORPUS						
alice29.txt	152089	192	2357	1371	12.28	7.14
asyoulik.txt	125179	127	1727	1059	13.60	8.34
cp.html	24603	15	96	64	6.40	4.27
fields.c	11150	8	19	21	2.38	2.62
grammar.lsp	3721	0	2	3	div 0	div 0
kennedy.xls	1029744	1230	35962	13796	29.24	11.22
lcet10.txt	426754	522	16447	6449	31.51	12.35
plrabn12.txt	481861	606	19295	9304	31.84	15.35
ptt5	513216	696	5323	3705	7.65	5.32
sum	38240	34	187	99	5.50	2.91
xargs.1	4227	2	6	2	3.00	1.00
LARGE CORPUS						
bible.txt	4047392	5168	332777	116260	64.39	22.50
E.coli	4638690	6307	337196	151540	53.46	24.03
world192.txt	2473400	3089	187505	65213	60.70	21.11
PURDUE CORPUS						
All_Up_1M.fasta	1001002	1200	55389	23982	46.16	19.98
All_Up_400k.fasta	399615	481	13294	6698	27.64	13.93
Helden_All.fasta	112507	122	1363	933	11.17	7.65
Helden_CGN.fasta	32871	34	232	178	6.82	5.24
Spor_All_2x.fasta	444906	57	34	44	0.60	0.77
Spor_All.fasta	222453	250	3314	2140	13.26	8.56
Spor_EarlyI.fasta	31039	26	220	190	8.46	7.31
Spor_EarlyII.fasta	25008	15	166	121	11.07	8.07
Spor_Middle.fasta	54325	62	506	396	8.16	6.39

Maximal Compression

sequence	size (chars.)	update	K & S	L & S	ratio K & S	ratio L & S
CANTERBURY CORPUS						
alice29.txt	152089	269	1091	510	4.06	1.90
asyoulik.txt	125179	182	866	405	4.76	2.23
cp.html	24603	18	55	40	3.06	2.22
fields.c	11150	3	18	6	6.00	2.00
grammar.lsp	3721	0	1	0	div 0	div 0
kennedy.xls	1029744	1541	4871	1671	3.16	1.08
lcet10.txt	426754	749	5815	2259	7.76	3.02
plrabn12.txt	481861	887	7841	2911	8.84	3.28
ptt5	513216	1900	842	369	0.44	0.19
sum	38240	28	82	48	2.93	1.71
xargs.1	4227	2	4	2	2.00	1.00
LARGE CORPUS						
bible.txt	4047392	10285	158048	38038	15.37	3.70
E.coli	4638690	14808	140788	31189	9.51	2.11
world192.txt	2473400	5573	90738	25276	16.28	4.54
PURDUE CORPUS						
All_Up_1M.fasta	1001002	2350	14109	4167	6.00	1.77
All_Up_400k.fasta	399615	884	2758	1129	3.12	1.28
Helden_All.fasta	112507	165	382	191	2.32	1.16
Helden_CGN.fasta	32871	31	55	50	2.89	2.63
Spor_All_2x.fasta	444906	61	35	71	0.57	1.16
Spor_All.fasta	222453	413	775	401	1.88	0.97
Spor_EarlyI.fasta	31039	25	56	47	2.24	1.88
Spor_EarlyII.fasta	25008	33	60	39	1.82	1.18
Spor_Middle.fasta	54325	73	117	66	1.60	0.90

Left Context Tree

Virtual Tree
If ω is subsequence of s:
then the left context tree of ω is defined:

- the nodes are subsequences of s
- the root is ω
- each node v has as childrens the nodes $av (a \in \Sigma)$ if av is a subsequence of s

An equivalent representation takes for each node v the interval over the suffix array such that all suffixes that start with a occurrence of v are contained in it.

Complications with lcp update